

⑫ 公開特許公報 (A)

平2-266465

⑬ Int. Cl. 5

G 06 F 15/30
G 09 C 1/00

識別記号

340

庁内整理番号

6798-5B
7343-5B

⑭ 公開 平成2年(1990)10月31日

審査請求 未請求 請求項の数 3 (全9頁)

⑮ 発明の名称 認証方式

⑯ 特 願 平1-87272

⑯ 出 願 平1(1989)4月5日

⑰ 発明者 岡本 龍明 東京都千代田区内幸町1丁目1番6号 日本電信電話株式会社内

⑰ 発明者 太田 和夫 東京都千代田区内幸町1丁目1番6号 日本電信電話株式会社内

⑰ 出願人 日本電信電話株式会社 東京都千代田区内幸町1丁目1番6号

⑰ 代理人 弁理士 草野 順

明細書

信し、

1. 発明の名称

認証方式

2. 特許請求の範囲

(1) 通信相手の身元を確認する利用者の認証方式において、

証明者Aと被検証者Bと検証者Cから構成されたシステムで、

証明者Aは初期応答文発生器と証明器を備え、被検証者Bは乱数発生器、初期応答文攪乱器、問い合わせ文攪乱器と乱数成分除去器を備え、検証者Cは検査器を備え、

証明者Aは、初期応答文発生器を用いて生成した初期応答文 x' を個人識別情報IDと共に被検証者Bに送信し、

被検証者Bは、証明者Aから受信した初期応答文 x' と乱数発生器を用いて生成した乱数成分を初期応答文攪乱器に入力して初期応答文 x'' を作成して受信したIDと共に検証者Cに送信し、

検証者Cは、被検証者Bに問い合わせ文 β を送

被検証者Bは、Cから受信した問い合わせ文 β と先に生成した乱数成分を問い合わせ文攪乱器に入力して問い合わせ文 β' を作成して証明者Aに送信し、

証明者Aは、初期応答文 x' と問い合わせ文 β' に対応した応答文 z を、IDに対した関係式 $s_j^2 \bmod N = f(ID, j)$ をみたす秘密情報 s_j を用いて動作する証明器を用いて生成して被検証者Bに送り返し(ここで、整数Nと関数fは公開情報)、

被検証者Bは応答文 z とIDを乱数成分除去器に入力して乱数成分の影響を取り除いて応答文 z' を求め、その値を検証者Cに送信し、

検証者Cは応答文 z' とIDを検査器に入力して z' が先に受信した初期応答文 x'' と先に送信した問い合わせ文 β に対する正しい応答になっていることを検査して、

被検証者Bが乱数成分を秘密にすることで、被検証者Bと証明者A間で通信される x' , β' ,

z と、検証者 C と被検証者 B 間で通信される x'' 、 β 、 z' の対応関係を秘密にできることを特徴とする利用者の認証方式。

(2) 請求項(1)に記載の手順を繰り返して、安全性を向上する利用者の認証方式。

(3) 通信文の正当性を確認するメッセージの認証方式において、

証明者 A と被検証者 B と検証者 C から構成されたシステムで、

証明者 A は初期応答文発生器と証明器を備え、被検証者 B は乱数発生器、初期応答文攪乱器、問い合わせ文発生器と乱数成文除去器を備え、検証者 C は検査器を備え、

証明者 A は、初期応答文発生器を用いて生成した初期応答文 x' を個人識別情報 ID と共に被検証者 B に送信し、

被検証者 B は、証明者 A から受信した初期応答文 x' と ID と乱数発生器を用いて生成した乱数成分とを初期応答文攪乱器に入力して初期応答文 x'' を作成し、その初期応答文 x'' と署名対象の

メッセージ m を問い合わせ文発生器に入力して問い合わせ文 β と β' を作成して β' を証明者 A に送信し、

証明者 A は、先に送信した初期応答文 x' と受信した問い合わせ文 β' に対応した応答文 z を、ID に対した関係式 $s_j z \bmod N = f(ID, j)$ をみたす秘密情報 s_j を用いて動作する証明器を用いて生成して被検証者 B に送り返し（ここで、整数 N と関数 f は公開情報）、

被検証者 B は応答文 z と ID と共に生成した乱数成分と問い合わせ文 β を乱数成分除去器に入力して乱数成分の影響を取り除いてメッセージ m に対応した値 z' を求め、 z' を m、ID、 β と共に検証者 C に送信し、

検証者 C は z' とメッセージ m と問い合わせ文 β と ID を検査器に入力して β と z' が m に対する正しい署名になっていることを検査して、

被検証者 B が乱数成分を秘密にすることで、被検証者 B と証明者 A 間で通信される x'' 、 β' 、 z と、検証者 C と被検証者 B 間で通信される m、

β 、 z' の対応関係を秘密にできることを特徴とするメッセージの認証方式。

3. 発明の詳細な説明

「産業上の利用分野」

この発明は、電気通信システムで電子資金移動等を行う場合に、消費者のプライバシーを保護できる通信プロトコルを実現できる認証方式である。

「従来の技術」

電気通信システムを用いた電子資金移動や IC カードを用いた決済が普及している。また、現金の代替手段としての汎用プリペイドカードの利用法や、電子財布の使用法が研究されている。このとき、資金の流れが特定の組織に管理されると、消費者の消費動向等の個人情報がその組織に蓄積され、プライバシー保護の観点から問題となる。

この問題の解決策として、暗号技術を用いて、資金移動の追跡を不可能とする安全な資金移動方式がある。例えば、David Chaum : "Security without Identification : Transaction systems to make Big Brother Obsolete", Communication

of the ACM, October 1985, Vol. 28 No. 10

Chaum の方式の概要は以下の通りである。

消費者（被検証者：B）が金額等の取引内容を含んだ文書 m を乱数で攪乱して通信文 z を作成して、z を銀行（証明者：A）に送信する。銀行 A は、消費者 B の正当性を認証した後にその消費者の口座から金額を引き落とし、金額に対応した署名を z に施して、署名付き通信文 z' を消費者 B に送り返す。消費者 B は、 z' から乱数の影響を取り除いて、m に署名を施した値として m' を求め、これを現金にかわる手段として商店（検証者：C）へ支払う。検証者 C は、 m' が銀行 A によって署名されていることを確認して、 m' がある金額の価値があると判断する。C は後日 m' を銀行 A に提出することによって、対応する金額を受け取る。すなわち、 m' は金券としての機能を備えている。

ここで、z は m に乱数が付加されているので、銀行および第三者は z から m を推定できないし、また、銀行と商店が結託しても m' と z の対応を

知ることができない。従って誰が m' を発行したかを知ることができない。これより、Chaum の方法では金券 m' の発行元（消費者）を推定できない（すなわち、追跡不可能）ので、消費者の消費動向等のプライバシーを守ることができる。

しかし、この方式は処理量の大きい RSA 暗号をベースにしているので、 z から z' を求めるための処理量の大きいことが問題となる（この例では銀行 A の処理量が大きくなる）。具体的には、RSA 暗号では、200 才同志の整数の乗法（ただし剰余計算を含む）が平均 768 回必要である。

ところで、高速な認証方式として Fiat と Shamir の方式がある（Fiat, A. and Shamir, A: "How to prove yourself : practical solutions to identification and signature problems", Proceedings of Crypto 86, Santa Barbara, August 1986, pp. 18-1-18-7）。

Fiat-Shamir 法では、処理量は、平均して $t (k+2)/2$ 回の乗算（ただし法 N における剰余計算を含む）で済む（k と t の意味は後述）、

特に、 $k = 5$ 、 $t = 4$ に選ぶことが推奨されているので、この場合には、Fiat-Shamir 法の乗算回数は 14 回となり、RSA 暗号による署名法に比較して処理量を大幅に削減できる。具体的には、 $14/768 = 0.02$ なので RSA 暗号に比べて約 2 % の処理量で実現できる。

Fiat-Shamir 法の概要は、以下の通りである。

信頼できるセンタが、個人識別情報として ID を用いる利用者に対して、次の手順で k 個の秘密情報 s_j ($1 \leq j \leq k$) を生成する（k は安全性を定めるパラメータであり 1 以上の値）。ここで N は公開情報であり、秘密の素数 P と Q を用いて $N = P \times Q$ と表せる。また f は一方向性関数であり、公開されている。

step1 : 一方向性関数 f を用いて

$x_j = f(ID, j)$ ($1 \leq j \leq k$)
を計算する。

step2 : 各 x_j に対して N の素因数 P と Q を用いて

証明者 A は、検証者 C に対して、A が本物であることを、次の手順で証明する。

step1 : A が ID を C に送る。

step2 : C が $x_j = f(ID, j)$ ($1 \leq j \leq k$)
を計算する。

次に、 $i = 1, \dots, t$ について 3 ~ 6 のステップを繰り返す（t は安全性を定めるパラメータであり、1 以上の値）。

step3 : 亂数 r_i を生成して、

$x'^i = r_i^2 \pmod{N}$
を計算して、C に送る。

step4 : C が、0, 1 のビット列 (e_{i1}, \dots, e_{ik}) を生成して、A に送る。

step5 : A が署名文 z_i を

$z_i = r_i \prod_{e_{ij}=1} s_j \pmod{N}$
で生成して、C に送る。

step6 : C は、

$x'^i = z_i^2 \pmod{N}$
 $e_{ij}=1$

$$s_j = \sqrt{x_j} \pmod{N}$$

を計算する。すなわち、 $s_j^2 = x_j \pmod{N}$ となる。

注) Fiat-Shamir 法では、実際には $s_j = \sqrt{N} x_j$ としているが、上記のように s_j を定めても同様の議論が成り立つ。

step3 : 利用者に対して k 個の s_j を秘密に発行し、一方向性関数 f と合成数 N を公開する。

(mod N) における平方根の計算は、N の素因数 (P と Q) が分かっているときのみ実行できる。その方法は、例えば Rabin, M.O.: "Digitalized Signatures and Public-Key Functions as Intractable as Factorization", Tech. Rep. MIT/LCS/TR-212 MIT Lab. Comput. Sci. 1979 に示されている。平方根の計算装置の具体的な構成例は、公開鍵暗号システム（特願昭61-169350）に示されている。

利用者の認証方式は以下の通りである。

が成り立つことを検査する。

$$z_j \text{ の作り方より } z_j^2 \bmod \prod_{e_{i,j}=1} x_i = r_j^2 \bmod \prod_{e_{i,j}=1} x_i$$

$$(s_j^2 \times x_j^{-1}) = r_j^2 = x'_j \pmod{N}$$

であるから、t回の検査にすべて合格した場合、検証者CはAが本物であると認める。

このとき、検証者Cが、偽の証明者を本物のAと認めてしまう誤りの生じる確率は $1/2^{kt}$ である。ここで、kは証明者が秘密に管理する s_j の個数であり、tは通信文の通信回数を定めている。

以上では、利用者の認証方式について説明したが、メッセージの認証方式は上記の手順を次の様に変更して実現できる。

メッセージmと (x'_1, \dots, x'_t) に一方向性関数fを施して得た $f(m, x'_1, \dots, x'_t)$ の先頭の $k \times t$ ビットを上記手順のビット列 $(e_{i,j})$ とみなして、署名文として、(ID, m, $(e_{i,j}), z_1, \dots, z_t$)を署名つき通信文として検証者に送信する。

このようにFiat-Shamir法は高速な認証方式で

証者C(300)が通信回線を介して接続しており、利用者の認証方式(図(a))とメッセージの認証方式(図(b))を実現するための交信例を表している。以下では、先ず、AがBの身元を確認したことをCに対して証明する利用者の認証方式を示し、その後に、BがAの力を借りてメッセージmに署名するメッセージの認証方式について説明する。

第1図の(a)では、A-B間とB-C間でそれぞれFiat-Shamir法の利用者認証法を採用し、2つのFiat-Shamir法を対応づける情報をBにおいて秘密にすることで、追跡不可能な利用者の認証処理を実現する。

Fiat-Shamir法の場合と同様に、信頼できるセンタが、合成数Nと一方向性関数fを公開し、さらに証明者Aの識別情報IDに対応する秘密情報 s_j を計算して、 s_j をAに配達する。ここで、 s_j は、 $s_j^2 \bmod N = x_j = f(ID, j)$ をみたすことに注意。

A(100)の概略を第2図に、B(200)

あるが、現在までのところFiat-Shamir法を用いた追跡不可能な認証方式は提案されていない。

「発明が解決しようとする課題」

この発明の目的は、システム設計者が処理速度を考慮して安全性のパラメータを選択できるようにして、従来方式よりも高速な追跡不可能な認証方式を提供することにある。

「課題を解決するための手段」

この発明では、処理量を削減するために、問い合わせ文と応答文を用いるFiat-Shamir法をベースにして、高速な認証処理を実現する。また、第三者にA-B間とB-C間で通信されるデータの対応関係を隠して、追跡不可能とするために、Bが問い合わせ文の対応関係と応答文の対応関係を乱数によって与え、その乱数を秘密にする。これによって、この発明では、追跡不可能な認証処理を、従来より少ない処理量で実現する。

「実施例」

第1図は、この発明の原理図である。第1図は証明者A(100)と被検証者B(200)と検

の概略を第3図に、C(300)の概略を第4図にそれぞれ示す。

証明者Aは、被検証者Bの正当性を、検証者Cに対して、次の手順で証明する。

step1 : AがIDをBとCに送る。

step2 : BとCは、それぞれ一方向性関数計算器205, 305を用いて $x_j = f(ID, j)$ を計算する。

次に、3~6のステップをt回繰り返す。t=1のときが特許請求範囲の請求項(1)に対応し、 $t > 1$ のときが請求項(2)に対応する。

step3 : Aは初期応答文発生器110を用いて初期応答文 x' を発生してBに送る。

例えば初期応答文発生器110を乱数発生器111と剰余付き乗算器112で構成して、乱数発生器111を用いて乱数rを発生し、剰余付き乗算器112を用いて

$$x' = r^2 \pmod{N}$$

で x' を計算する。

剰余付き乗算の効率のよい計算方法は、例えば

池野、小山“現代暗号理論”電子通信学会、pp. 16-17、(1986)、に示されている。

step4 : Bは x' を受信すると、乱数発生器210と初期応答文搅乱器215を用いて、乱数発生器210で発生したk個の0,1のビット (e_j) と乱数uを x' と先に生成したxと共に初期応答文搅乱器215に入力し、搅乱された初期応答文 x'' を計算してCに送る。

例えば初期応答文搅乱器215を剩余付き乗算器として構成し、受信した初期応答文 x' と x'' と (e_j) とuから

$$x'' = x' \times u \times \prod_{e_j=1} x_j \pmod{N}$$

で x'' を計算する。

step5 : Cは x'' を受信すると、 x'' を秘密情報格納器310に格納した後に、乱数発生器320を用いて、k個の0,1のビット (β_j) を生成して $\beta = (\beta_1, \dots, \beta_k)$ を問い合わせ文としてBに送る。

で z を計算する。

step8 : Bは z を受信すると、 z と先に生成した (x_j) と (e_j) とuを乱数成分除去器230に入力して、応答文 z' を計算してCに送る。

例えば乱数成分除去器230を、条件判定器231と剩余付き乗算器232で構成し、

$$z' = u \times z \times \prod_{c_j=1} x_j \pmod{N}$$

ただし、 $c_j = \beta_j$ and e_j

を計算する。

step9 : Cは z' を受信すると、検査器330を用いて z' の正当性を検査する。

例えば検査器330を、剩余付き乗算器331と比較器332で構成し、秘密情報格納器310から引き継いだ x'' と一方向性関数計算器305から引き継いだ x_j と乱数発生器320から引き継いだ β に対して

$$x'' = z'^z \times \prod_{\beta_j=1} x_j \pmod{N}$$

step6 : Bは β を受信すると、 β と先に生成した (e_j) を問い合わせ文搅乱器220に入力して、搅乱された問い合わせ文 $\beta' = (\beta'_1, \dots, \beta'_k)$ を計算してAに送る。例えば問い合わせ文搅乱器220を排他的論理和計算器として構成して、

$$\beta'_j = \beta_j \oplus e_j$$

を計算する。

step7 : Aは β' を受信すると、先に生成した乱数rと受信した問い合わせ文 β' を証明器120に入力して、応答文zを計算してBに送る。

例えば証明器120を、秘密情報格納器121と剩余付き乗算器122で構成し、秘密情報格納器121から秘密情報 (s_j) を読み出して、初期応答文発生器110から引き継いだrと受信した β' を剩余付き乗算器122に入力して

$$z = r \times \prod_{\beta'_j=1} s_j \pmod{N}$$

が成立するかを検査する。

ここではt回の問い合わせー応答のやりとりを順次行う例を示したが、問い合わせー応答のやりとりを同時にやってもよい。

次に、第1図の(b)を用いて、BがAの力を借りてメッセージmに署名するメッセージの認証方式について説明する。

A-B間ではFiat-Shamir法の利用者認証法を、B-C間ではFiat-Shamir法のメッセージ認証法を採用する。2つの認証法を対応づける情報をBにおいて秘密にすることで、追跡不可能なメッセージの認証処理を実現する。

Fiat-Shamir法と同様に、信頼できるセンタが、合成数Nと一方向性関数fを公開し、さらに、証明者Aの識別情報IDに対応する秘密情報 (s_j) を計算して、 (s_j) をAに配達する。

A(100)の概略を第2図に、B(200)の概略を第5図に、C(300)の概略を第6図にそれぞれ示す。

Bは、Aの力を借りて、次の手順で文書mに署

名する。

step1 : A が ID を B と C に送る。

step2 : B と C は、それぞれ一方指向性関数計算器 205, 305 を用いて $x_j = f(ID, j)$ を計算する。

step3 : A は初期応答文発生器 110 を用いて t 個の初期応答文 x'_i ($i = 1, 2, \dots, t$) からなる x' を計算して B に送る。

例えば初期応答文発生器 110 を、乱数発生器 111 と剰余付き乗算器 112 で構成し、乱数発生器 111 を用いて t 個の r_i を発生し、剰余付き乗算器 112 を用いて

$$x'_i = r_i z \pmod{N} \quad (i = 1, 2, \dots, t)$$

で、t 個の x'_i を計算する。

step4 : B は x' を受信すると、乱数発生器 210 を用いて t 組の k ビット $\{e_{ij}\}$ と乱数 u_i のペアを発生し、その値を受信した t 個の x'_i と先に生成した $\{x_j\}$ と共に初期応答文搅乱器 215 に入力し、t 個の搅乱さ

$$\beta'_{ij} = \beta_{ij} \oplus e_{ij} \quad (i = 1, 2, \dots, t, j = 1, 2, \dots, k)$$

で、 $\beta = \{\beta_{ij}\}$ と $\beta' = \{\beta'_{ij}\}$ を求める。

step5 : A は β' を受信すると、証明器 120 を用いて、先に発生した乱数 r_i と受信した問い合わせ文 β' から、応答文 z を計算して B に送る。

例えば証明器 120 を、秘密情報格納器 121 と剰余付き乗算器 122 で構成し、秘密情報格納器 121 から秘密情報 $\{s_j\}$ を読み出し、初期応答文発生器 110 から引き継いだ $\{r_i\}$ と受信した β' を剰余付き乗算器 122 に入力して

$$z_i = r_i \times \prod s_j \pmod{N} \quad \beta'_{ij} = 1$$

$$(i = 1, 2, \dots, t)$$

で計算した z_i を用いて、 $z = (z_1, \dots, z_t)$ を求める。

step7 : B は z を受信すると、 z と先に生成し

れた初期応答文 x'_i を計算して $x'' = (x''_1, \dots, x''_t)$ を問い合わせ文発生器 250 に引き継ぐ。

例えば初期応答文搅乱器 215 を剰余付き乗算器で構成し、乱数発生器 210 が生成した t 組の $\{e_{ij}\}$ と u_i 、受信した t 個の初期応答文 x'_i と x''_i を初期応答文搅乱器 215 に入力して

$$x''_i = x'_i \times u_i^2 \times \prod x_j \pmod{N} \quad e_{ij} = 1$$

$$(i = 1, 2, \dots, t)$$

で t 個の x''_i を計算する。

step8 : B は、メッセージ m と t 個の x''_i を問い合わせ文 β と β' を作成して β' を A に送信し、 β を乱数成分除去器 260 に引き継ぐ。

例えば、問い合わせ文発生器 250 を一方指向性関数計算器 251 と排他的論理和計算器 252 で構成して、

$$\{\beta_{ij}\} = f(m, x''_1, \dots, x''_t)$$

た $\{x_j\}$ と t 組の $\{\{e_{ij}\}, u_i\}$ を乱数成分除去器 260 に入力して、応答文 z' を計算して、 β 、m と共に C に送る。

例えば乱数成分除去器 260 を、条件判定器 261 と剰余付き乗算器 262 で構成し、

$$z'_{ij} = u_i \times z_i \times \prod x_j \pmod{N} \quad c_{ij} = 1$$

$$(i = 1, 2, \dots, t)$$

ただし、 $c_{ij} = \beta_{ij}$ and e_{ij}

で計算した z'_{ij} を用いて、 $z' = (z'_1, \dots, z'_t)$ を求める。

step9 : C は m、 β 、 z' を受信すると、検査器 340 を用いて m、 β 、 z' の正当性を検査する。

例えば検査器 340 を、剰余付き乗算器 341 と一方指向性関数計算器 342 と比較器 343 で構成し、

$$x''_i = z'_{ij} \times \prod x_j \pmod{N} \quad \beta_{ij} = 1$$

で $x'' = (x''_1, \dots, x''_t)$ を求めて、

$$\{\beta_{ij}\} = f(m, x^*)$$

が成立するかを検査する。

以上では、Fiat-Shamir法をベースにした追跡不可能な認証方式について説明した。Fiat-Shamir法は、Nの素因数分解が困難な場合に $(\bmod N)$ での平方根の計算が困難なことに基づいている。離散対数問題等の困難性を利用して認証法をベースにしても、同様の議論が成り立つ。離散対数問題等に基づく認証法については、例えば M.Tompa & H.Woll, "Random Self-Reducibility and Zero Knowledge Interactive Proofs of Possession of Information," FOCS, pp472-482(1987) や岡本、太田, "零知識証明問題の不正使用法とその対策及び応用について" (1988年暗号と情報セキュリティシンポジウムワークショップ) に示されている。

「発明の効果」

この発明では、Fiat-Shamir法をベースにしたので、高速な認証処理を実現できる。

また、Bが問い合わせ文の対応関係と応答文の

対応関係を秘密の乱数で与えておりその値を秘密にすると、A-B間とB-C間で通信されるデータの対応関係を隠すことができる。すなわち、利用者の認証処理においては、AがBの身元を保障していることを、Bの身元を明かさずに、Cに証明できる。メッセージの認証処理においては、Bはメッセージmの内容を知られることなしにAに署名させることができる。その結果として、AとCが結託しても、Bの身元は明らかとならず、Bがmを送信したこと検出できない。すなわち、追跡不可能な認証処理を実現できる。以上より、この発明では、従来方式より高速に追跡不可能な認証処理を実現できる。

証明者Aと検証者Cが結託しても、被認証者が誰であるかを判断したり、メッセージmの送信者が誰であるかを判断したりできないことは、この発明の方式が計算量理論の理論的な研究成果である零知識対話型証明システム性や非転移性をみたすことによって保障できる。

零知識対話型証明システム性および非転移性に

については、例えば Feige,U., Fiat,A. and Shamir, A, "Zero Knowledge Proofs of Identity" Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, pp.210-217. を参照。

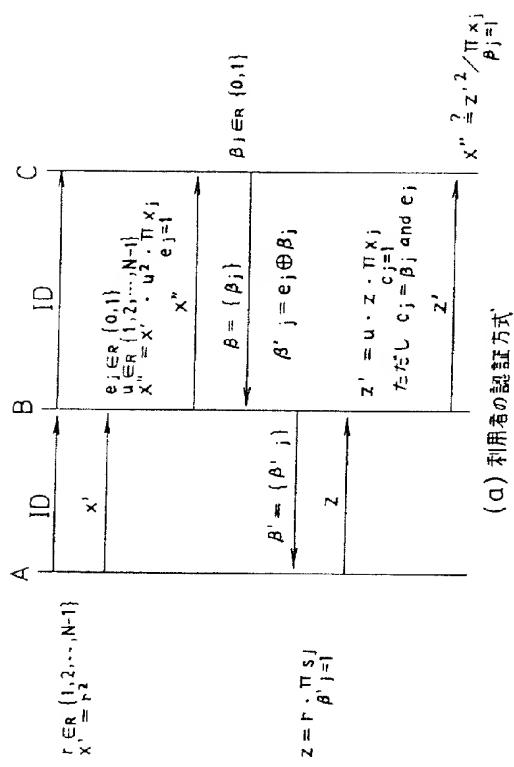
4. 図面の簡単な説明

第1図はこの発明の実施例の交信例を示す図、第2図は証明者Aのブロック図、第3図は利用者の認証方式における被検証者Bのブロック図、第4図は利用者の認証方式における検証者Cのブロック図、第5図はメッセージの認証方式における被検証者Bのブロック図、第6図はメッセージの認証方式における検証者Cのブロック図である。

特許出願人 日本電信電話株式会社

代理人 草野 韶

卷一



(a) 利用者の認証方式

A	$1D$	x	x'	$\beta' = \beta'_{ij}$	z	$m, z', \{\beta'_{ij}\}$
B	$1D$				$z' = (z'_1, \dots, z'_t)$	
C					$z'_i = u_i \cdot z_i : \prod_{j=1}^t x_j$	
					$\beta'_{ij} = \beta_{ij} \oplus e_{ij}$	
					$\beta'_{ij} = (z'_1, \dots, z'_t)$	
					$x''_i = u_i^2 \cdot \prod_{j=1}^{N-1} x_j$	
					$x''_i = \{x''_1, \dots, x''_{N-1}\}$	
					$e_{ij} \in R^{\{0,1\}}$	
					$u_i \in R^{\{1,2, \dots, N-1\}}$	
					$x'_i = (x'_1, \dots, x'_t)$	
					$x'_i \in R^{\{1,2, \dots, N-1\}}$	
					$\beta'_{ij} \in R^{\{0,1\}}$	

(b) メッセージの認証方式

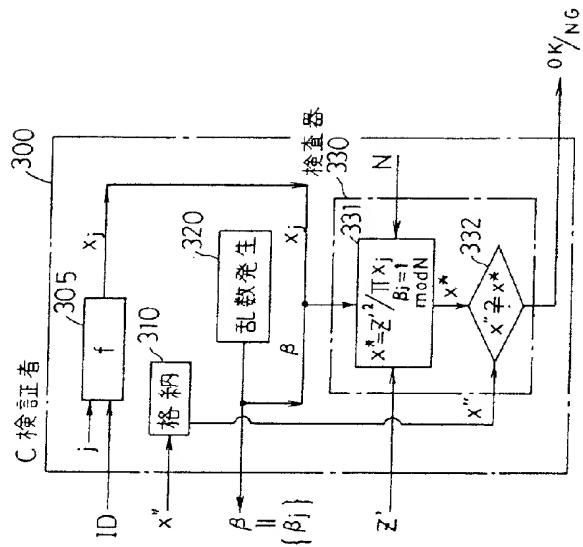
九三

第 5 回

The diagram illustrates a challenge-response protocol between a verifier (B) and a prover (200). The verifier sends a challenge x_i to the prover. The prover performs the following steps:

- Generates a random number e_{ij} (210).
- Computes $x_i^* = u_i \cdot x_i \cdot \prod x_j \pmod{N}$ (215).
- Computes $e_{ij}^* = e_{ij} \cdot u_i \pmod{N}$ (250).
- Computes $e_{ij}^* \oplus \beta_{ij}$ (250).
- Generates a witness $m, (x_i^*, e_{ij}^*)$ (251).
- Computes $\beta = \{\beta_{ij}\}$ (250).
- Computes $\beta = \{\beta_{ij}\}$ (250).
- Sends (m, β, e_{ij}^*) to the verifier.
- Computes $e_{ij} = e_{ij} \cdot u_i \pmod{N}$ (210).
- Computes $C_{ij} = \beta_{ij} \cdot e_{ij} \pmod{N}$ (261).
- Computes $Z'_i = u_i \cdot Z_i \cdot \prod x_j \pmod{N}$ (262).
- Computes $Z' = (Z'_1, \dots, Z'_t)$ (205).
- Computes $Z' = (Z'_1, \dots, Z'_t)$ (205).

ガ 4 図



ガ 6 図

